

Dipesh Pradhan

Cross-platform Mobile and Tablet Application

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

20 November 2011

Author Title	Dipesh Pradhan Cross-platform mobile and tablet application
Number of Pages Date	50 pages + 5 appendices 20 November 2011
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructors	Kari Aaltonen, Senior Lecturer Petri Ebeling, Chief Technology Officer
<p>The goal of this project was to develop a cross-platform mobile and tablet application and study it further to make it secure and cost efficient. The possibility to offer native application features was also analyzed. This project was carried out for Process Vision Oy to deploy the desktop software developed there for mobile devices and tablets.</p> <p>The application was developed as a web application to make it cross-platform. A mobile framework called jQuery Mobile was used along with markup language, style sheets and scripting language to build the web application. A custom JavaScript file was used for form validation and selecting different dates. Similarly, web service was studied to secure the application and to run it across various clients.</p> <p>Furthermore, the developed web application was tested with real devices and emulators using native and additional browsers on various operating systems. The results showed that the application performed well on different platforms and only a few features were not supported by all the browsers.</p> <p>Finally, the application was demonstrated to the Process Vision staff and the instructors of thesis. Positive feedback was received as a result of the evaluation. Future developers can use this project to learn more about building cross-platform application.</p>	
Keywords	Cross-platform, jQuery Mobile, web service, PhoneGap

Contents

1	Introduction	1
2	Platforms	2
2.1	Apple's Mobile Operating System	3
2.2	Android	5
2.3	Windows Phone	7
2.4	BlackBerry and Bada	8
3	Web Application Development Tools	11
3.1	Markup Language and Style Sheets	13
3.2	Frameworks	17
3.3	Web Services	21
4	Methods and Materials	24
4.1	Development Tools	24
4.2	Overall Design	25
5	Application Description	29
5.1	Architecture	29
5.2	User Interface	31
5.3	Web Service Recommendation	37
6	Results	38
7	Discussion	41
8	Conclusion	44
	References	45

Appendices

Appendix 1. Ajax Navigation

Appendix 2. JavaScript Validation and Popup Page

Appendix 3. Custom Templates and Select Menu

1 Introduction

Application development for different mobile and tablet platforms is very expensive and time consuming. The same application has to be developed multiple times in several programming languages using platform specific development tools to cover all the platforms. Similarly, the cost to use development tools and application revenue taken by the operating systems for selling the applications through their stores results in a lot of wasted revenues for companies. Hence, there is a need to develop cross-platform applications capable of running across all mobile and tablet platforms, which is more cost effective and less labor-intensive.

The company, Process Vision Oy, in which I carried out my thesis project, had similar problems. They wanted to deploy a desktop application developed there called eGeneris for mobiles and tablets as well. Initially, they have developed software called Generis. It is an IT system solution for energy companies covering the majority of business processes executed by energy companies from meter management to billing and trading. In turn, eGeneris is an extranet web portal offering access to a selection of data and business use cases supported by Generis. However, developing and maintaining multiple versions of the same application is very expensive as stated earlier, and thus they can only support a single application.

Since only a single application can be created and supported by the company, it results in marginalization of the customers who use other operating systems. Thus, this has led me to research into a cross-platform, by the use of which eGeneris can be developed to run on mobiles as well as tablets for a wide variety of clients using various operating systems.

The goal of this project is to develop a cross-platform energy market application accessible with mobile devices and tablets irrespective of their platforms by selecting a suitable technology. After the application development, the application was researched further to make it secure and cost efficient using the best web service framework. The possibilities to offer native features were also studied as they would be useful in the future if the company decided to increase the number of features of the application.

2 Platforms

Smartphones have extra features such as cameras, the Global Positioning System (GPS), accelerometers and other sensors which can be used to add additional features in the applications. With those features, the applications can, for example, tell the location of the user, take picture and access contact details, use distinct motions such as shake and vibrate to enable different actions in the application, and other additional features. Some of these features are also supported by tablets. Nevertheless, they are not supported by desktops. This has led to the need to develop separate applications for mobile devices and tablets by abandoning the traditional way of developing applications.

Additionally, there are many different operating systems available for smartphones and tablets. Each operating system (OS) is optimized to work best for a particular set of devices. The human nature also prompts people to believe that it is possible to develop new and better systems all the time. This led to the development of several different operating systems for desktop computers in the 1980s and 1990s; however, only a few still exist. Today, the case is similar for mobiles and tablets. There used to be many different operating systems, but now companies are abandoning their OS for some other OSs or quitting the business altogether [7].

To develop applications on different platforms, developers need to use platform specific tools, Application Programming Interfaces (API), Software Development Kits (SDK) and platform specific languages. It is possible to use a different language than the one used in the platform; however, there are limitations and drawbacks. For example, when developers use Java to build an application, several native APIs for accessing device capabilities will be unavailable. [1, 1-5]

Finally, after an application has been developed, developers can distribute it using the application store of different operating systems. All the various platforms have their own application store where users can download free as well as paid applications for their mobile devices. For example, Apple has App Store while Google has Android Market. Furthermore, there is Windows Phone Marketplace for Microsoft devices and BlackBerry App World for Research in Motion (RIM) devices. Other companies have

also followed this trend of creating an application store for their operating systems. For example, Nokia has OVI Store while Samsung has introduced Samsung Apps Store for its Bada smartphone platform along with Android and Windows Mobile devices. Table 1 below shows comparison between various operating systems.

Table 1. Comparison between different operating systems. Data gathered from [1, 4].

OS	Android	iOS	WP 7	BlackBerry	WebOS
Kernel	Linux kernel	Mach kernel	Windows CE 6		Linux kernel
Application framework	Android framework	Cocoa Touch	SilverLight And XNA	BlackBerry API UI	webOS Enyo
Supported CPU architecture	ARM, x86	ARM	ARM		ARM
IDE	Eclipse	Xcode	Visual Studio	Eclipse	Eclipse
SDK	Android SDK	iOS SDK on Mac OSX	Windows Phone SDK	BlackBerry Java SDK	webOS SDK
Language	Java	Objective-C	C# and .NET	Java	JavaScript

As can be seen in table 1, the operating systems have a lot of differences in their structure. Developers need to have a good understanding of each operating system to develop applications for them. It is also essential to know how to develop and distribute applications for different platforms. Thus, a general overview will be given of each operating system, which will increase understanding of their strong and weak points.

2.1 Apple's Mobile Operating System

Apple's mobile operating system is called iOS, and it has been further extended by Apple to support devices such as iPod touch and iPad. Apple developed iOS from the Macintosh (Mac) OS X which is its desktop OS for desktop computers. Both the Mac OS X and iOS operating systems were built using the C programming language. [2, 26] To develop applications, developers need to have an Intel based Mac computer with the operating system OS X v10.5.7 or later installed. iOS uses Objective C as the

programming language. Objective C was developed by modifying a standard C compiler to add the best features of the SmallTalk programming language [1, 17]. C and C++ can also be used to write applications on the iOS. However, since the graphical user interface and core framework libraries are written in objective C, developers use objective C for easy application development [3, 9].

Four different technology layers are present in iOS. The base layer is known as the core OS layer. On top of it, there is the core services layer and above it the media layer which contains the C application programming interfaces. Finally, the top layer is called the Cocoa Touch, and it is used to simplify the OS programming. Primary class libraries are provided by Cocoa Touch, and there are two Cocoa Touch frameworks needed by the developers: Foundation framework and UIKit framework. The foundation framework is related to standard programming topics while the UIKit framework is for the iPhone and iPad's device interface. [2, 6]

Furthermore, multitasking is supported by iOS, and there is a sequence of events between the application launch and termination. Different events occur at various part of the application's life cycle. The applications running there can be in one of the following states: not running, inactive, active, background and suspended. If the application has not been launched or it has been terminated, it belongs to the not running state. After the application is paused, it reaches the inactive state and when the user is interacting with the application, it reaches the active state. [4, 49-50]

Similarly, when the activity is executing in the background, it is in the background state. Finally, if the application is in the background but not executing any code, it is in the suspended state. [4, 50] The unique combination of software and hardware along with features such as multi touch in iPhone enable the developer to create good applications which can make maximum use of the hardware and software.

Apple has changed the mobile phone market by introducing the App Store. Consumer spending there has increased along with increase in the number of applications. Today, Apple's App Store has the greatest number of applications and maximum number of downloads. [1, 2] Nonetheless, Apple's application review process is very tough, and it takes days to weeks for the applications submitted by the developer to be displayed on

the App Store. This is due to the fact that the iOS is a closed integrated system meaning that the applications are regularly monitored and reviewed by Apple. This has helped Apple to control the quality of applications in the App Store, and Apple has also used it to create a monopoly on the market.

2.2 Android

The Android platform consists of an operating system, a number of higher-level libraries that simplify the task of communicating with the operating system and key applications. It is based on the Linux open source platform and Oracle's Java programming language. It was designed as a generic OS for smartphones and tablet personal computers which the Original Equipment Manufacturer could customize and sell to customers. Extensible Markup Language (XML) is used to define the basic components and the visual components in Android. The runtime in Android translates the Java and XML code written by developers to language understood by the operating system and devices. [5, 41]

Similarly, five major components are placed in four stacks in Android: applications, an application framework, and libraries along with Android runtime and Linux kernel. Applications are written using Java programming language. Developers have the same access rights as Google, and they can use the framework API to its maximum capacity to develop applications. A set of C/C++ libraries used by various components of the Android system are listed in the libraries, and these can be used by the developers through the Application Framework. [6]

Every application in Android runs in its own isolated environment since each application has its own processes, and each process has its own virtual machine [4, 47]. In Android, an application runs continuously until the application processes need more system resources. When the system needs additional resources, the Android system terminates the application. The Linux kernel provides the underlying functionality, such as threading and low-level memory management to the virtual machine [6]. Similar to App Store in iOS, Android Market is available in Android, which lets the users download content free of charge or by paying a fee. Compared to iOS, it is quite easy to submit applications in Android Market, and the applications are not reviewed there. This has

also led to an increased number of malware in the Android Market. Figure 1 illustrates the Android architecture and how the stacks are arranged there.

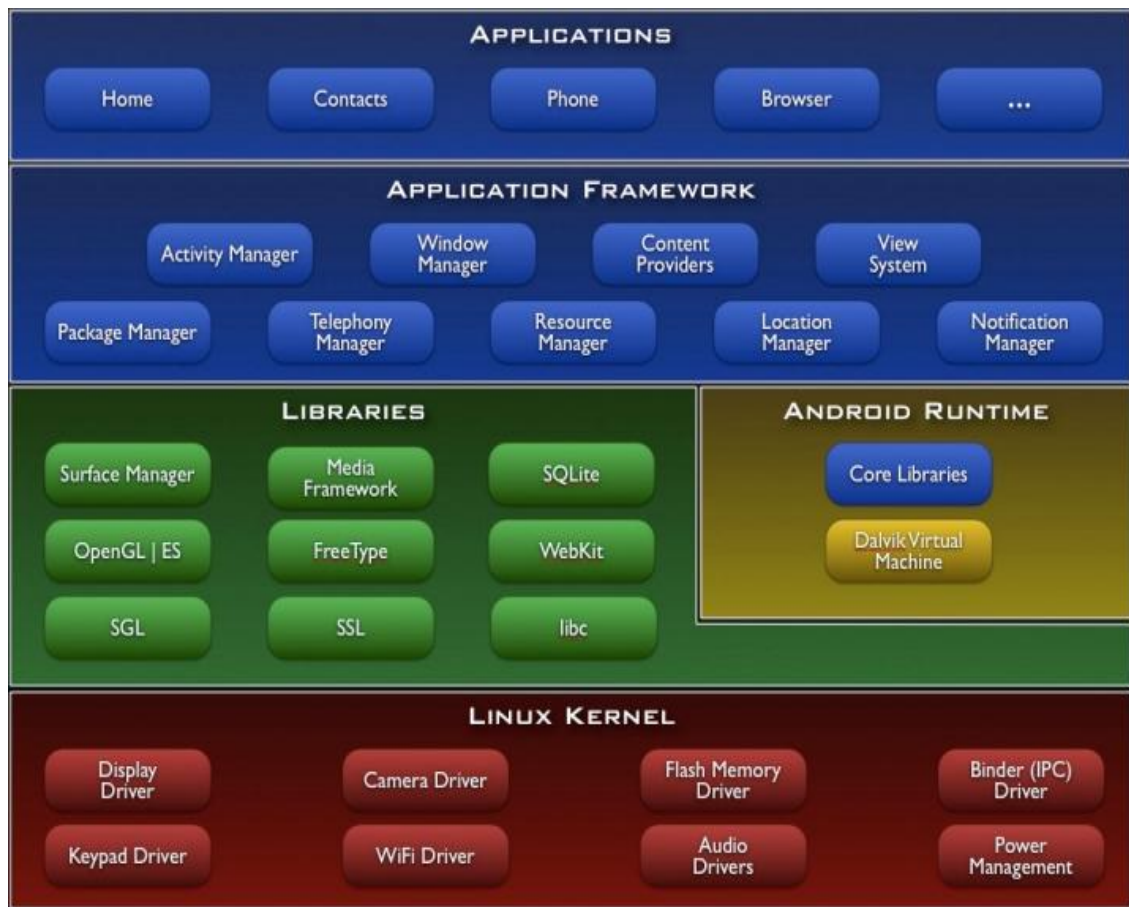


Figure 1. Android stack. Reprinted from Google [6].

As seen in figure 1, applications comprise the top level which is the first level of the stack. The Android OS is shipped with a set of core applications such as a search engine, an email client, a calendar, and maps. Application Framework is the second level of the stack. Different services and systems run underneath the applications, and these form the application framework. Libraries are the third level of the stack along with Android Runtime. The Linux kernel is the fourth level of the stack, and it is the core of the Android device software. The drivers and the hardware abstraction layer modules are all Linux based. The kernel acts as an abstraction layer between the hardware and the rest of the software stack. [6]

Furthermore, if the developers use the C/C++ code from another platform, they need to wrap the existing code into native libraries. The user interface in Android is

composed of different view components comprising of combined layouts. It is possible for the developers to inject native libraries to a Java application. When this is done, Java Native Interface (JNI) is used for the interaction between the Java code and the native code. [4, 11]

Additionally, an Android application is composed of activity, service, broadcast receivers and content providers. Each of them has a separate task and a distinct lifestyle. An application can have a number of activities working together, but each activity is independent of the other. The user interface of the smartphone screen is designed by the activities. The service runs in the background and does background processing such as playing music and processing game data. Broadcast receivers handle communication between various activities in the application. They are used for providing notifications and announcements. The content provider is used for data management and to maintain a collection of shared information data. [5, 81]

Finally, as the number of Android devices has increased in recent months, the Android Market has also increased to keep up with the trend. Android Market is known for its openness, and any application submitted by the developer appears immediately there. This has been both an advantage and disadvantage as it means that developers can immediately distribute their applications while the quality of applications in the Android Market has deteriorated due to lack of monitoring the applications.

2.3 Windows Phone

Windows Phone 7 is the latest version of Windows Phone. Microsoft launched Windows Phone 7 worldwide for mobile devices in 2010. Unlike the previous version of Windows Mobile which was available on devices with varying screen resolutions and hardware features, Windows Phone 7 Series uses a standardized set of hardware for easy application development. Similarly, the SDK for Windows Phone 7 utilizes .NET and Silverlight which have been used by developers for a long time. Silverlight and .NET are application frameworks that enable the developers to develop and run web applications. Microsoft has released a set of capabilities each Windows Phone 7 device must have to offer similar user experience across all Windows Phone 7 devices. [23, 4-5]

Unlike iOS and Android, Windows Phone 7 does not allow multi-tasking between third party applications. It only allows one application to run at the foreground at any given time. If the application is not present in the foreground, it is put in a dormant state by the operating system. The application remains in a dormant state as long as there is sufficient device memory available for the foreground application. A programming framework is provided for applications to manage their state to manage a good user interface. [26]

Similarly, for application state management, Windows Phone has four primary events: launching, activated, closing and deactivated. When the application is launched, the launching method is called. After the user returns to the dormant or tombstoned application, an activated event takes place. After the user presses the back button as many times as necessary to navigate backwards through the pages of the application and past the first page of the application, an application closing event occurs. With the application closing event, the user can exit the application, and the operating system removes any trace of that application from the RAM. An application deactivated event occurs after a different application runs on the foreground. However, unlike the closing event, the deactivated event gets tombstoned and can be restored at a later time. [26]

Furthermore, using the concept of tombstoning and the back button, two distinct user experiences can be made possible in a Windows Phone application when it becomes invisible to the user from the foreground. This can give the feeling that multiple applications are running there. Unlike with iOS and Android, users can navigate from the application to the browser or another application, and use the back button to return to the previous application using the Windows Phone operating system. The application submitted on the Windows Phone marketplace is tested both statically and automatically to ensure that the application is reliable, stable, performs well, does not interfere with device functionality, and is free from malicious bugs. [4, 12]

2.4 BlackBerry and Bada

In addition to the operating systems described above there are other operating systems. Research In Motion's (RIMs) popular BlackBerry and Samsung's Bada smartphone platform are worth mentioning in this regard because they are still used

widely. The future of other popular mobile operating systems such as Symbian, MeeGo and webOS do not look too bright. This is due to the fact that Nokia moved to Windows Phone and stopped supporting Symbian and MeeGo. Furthermore, Nokia is planning to end its support for Symbian by 2016 and has already started to develop its high end devices using Windows Phone OS. In addition to that, Hewlett-Packard (HP) has stopped the production of their webOS tablets due to limited customer interests and low sales. RIM is also facing similar challenges with declining sales. [7]

RIM developed BlackBerry, and it is used to run Java applications. Different types of applications can be built there. However, the most common type of applications in BlackBerry are browser based applications. BlackBerry widgets allow the developers to deploy web applications to devices by packaging them into a Java application. The BlackBerry platform allows browser applications to access GPS location information as well as use the special features. The web application can be packaged into a Java application and deployed to devices using BlackBerry widgets. [8, 2-3]

Furthermore, BlackBerry is mostly known for its security. Very few file types, mostly images and documents are supported there, and the data sent or received is encrypted to prevent snooping. For further protection, a signature process is implemented by RIM forcing developers to identify themselves and their applications if they want access to the core application or the OS. This has prevented the applications from doing anything harmful. Neither the messages nor the email or attachment viewer in BlackBerry support scripting so users are safe from receiving viruses through emails. Thus, with these security measures RIM has gained the trust of the US government and many of the Forbes top 500 enterprises in the financial and health sectors. [9, 17]

Bada is an open smartphone platform developed by Samsung for its mobile phones, and it originates from Samsung's proprietary platform first used in 2004. Service APIs such as social networking, friend lists, weather services, shopping and commerce are integrated in the Bada platform which can be used by any third party application. Developers can choose APIs from a small selection of chosen APIs to create simple powerful applications. The Bada applications are designed for the mass market of ordinary users who are using mid-range mobile devices which are neither too expensive nor too cheap. [10, 2-3]

Additionally, the C++ class library is available as a framework layer in Bada. A mobile kernel is present there along with two layers which control the device and access services. It is written in C++ on top of the C/C++ middleware. The Bada applications have access to native OS threading, and they run in a native OS process. The application also has integration for flash and WebKit support in its native APIs allowing inter-operation and multiple language code. [10, 2-4] There are a lot of different operating systems for smartphones. Figure 2 below shows the worldwide percentage of smartphone sales to the end users in the second quarter of 2011.

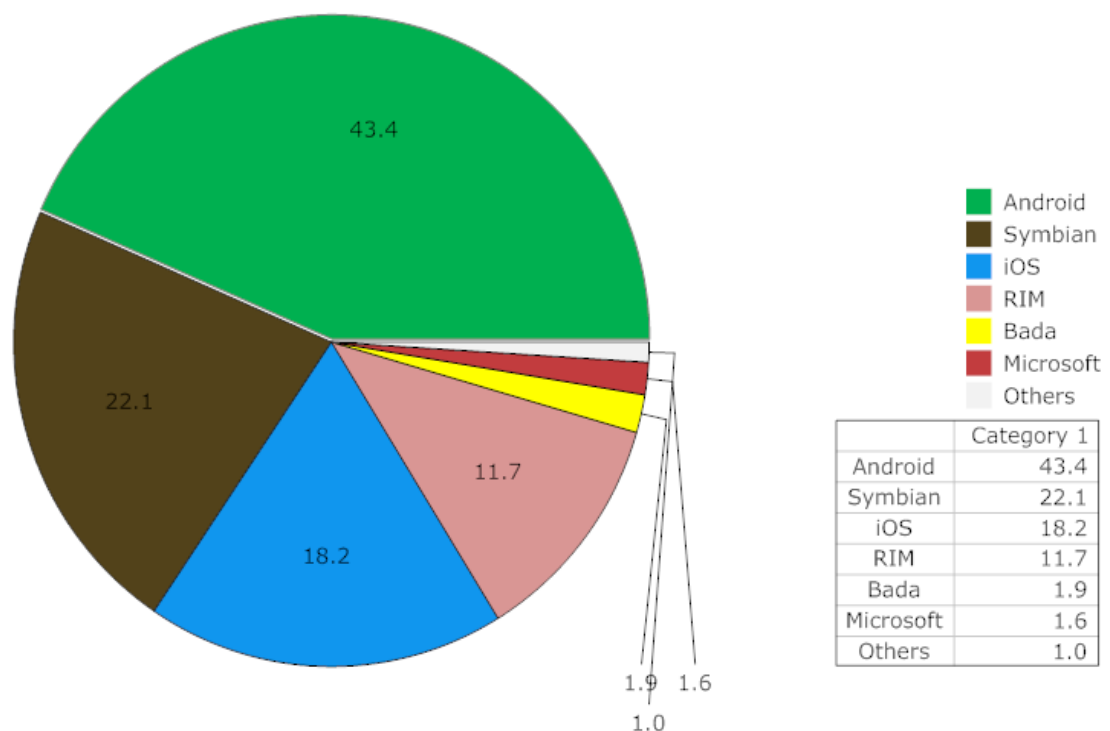


Figure 2. Worldwide smartphone sales to users in Q2 2011. Data gathered from Egham [11].

As can be seen in figure 2, Android was leading the smartphones' sale market in the second quarter of 2011. It had a lot of manufacturing companies' support and it dominated the smartphone market with varieties of smartphones. Although Nokia has moved to Windows Phone 7, Symbian has the second largest market share since Nokia has not yet abandoned it, and Nokia is in a transition phase from Symbian to Windows Phone. Symbian is doing particularly well in the Asian market. Apple's popular iOS holds the third position while RIM is at the fourth position with BlackBerry; however, RIM is also losing market share along with Nokia.

Furthermore, Samsung's own OS Bada is at the fifth position. This is due to the fact that Samsung's high end smartphones use Android and Windows Phone. Thus, Samsung is not focusing on just one platform. Microsoft is at the sixth position and at the bottom of the smartphone race with its Windows Phone OS. Nevertheless, there are a lot of manufacturing companies involved in Windows Phone, so the market share of Microsoft is bound to increase in the future. Finally, the other remaining operating systems make up for the remaining one percentage market share.

3 Web Application Development Tools

When the user opens any web application, the web browser allows the user to request any resources such as picture, documents and pdf files. The browser then sends this request to the web server which will either return the resource to the user or display an error message if the web server cannot find the resource. [12, 4] After that, the browser shows the user the result of the request. The resource can be anything and the server returns it if it is requested by the user. In simple words, this process is a synchronous activity where the web server waits for a request, and then returns the appropriate resource. The web client is the web browser application doing what the user asked it to do.

Additionally, the programmable web uses Hypertext Transfer Protocol (HTTP) as the transportation medium and XML is often used as the message format. The web server answers the client's request with a set of instructions written in the markup language. The markup language then tells the browser how to display the content to the user. An HTTP request is sent to the server by the web browser, and the server responds back with an HTTP response. Thus, the conversation between the web server and client is maintained. [12, 6]

Similarly, when the web site is used to complete an activity, interaction is either client sided or server sided, which means that information is processed on the client's computer or by the server itself after which it is sent to the client. A client side scripting language such as HTML, JavaScript, or CSS is run on the browser of user, and the user can view the source code. However, server side languages such as PHP, ASP

and JSP are executed on the web server, and the code to generate the web page is generated and returned to the web browser which then displays the output. Thus, the users cannot view the programming code of the server side scripting. JavaScript cannot store or retrieve data from the web server while it is not possible to initiate actions based on the user's mouse action or screen size using PHP. As a result, both the server side and client side scripting languages are essential for a web application.

Furthermore, the responsibilities for application behavior are divided between a remote web server and a local web browser. Browsers act as the door to the internet. The browser is used to handle the user interface so that users can have a good user interface, and the users get what they expect with a smooth flow of actions. The server on the other hand generates the web pages and stores data in the database. This helps the browser to be lightweight while all the intensive tasks are handled by the web server.

Mobile browser applications are web sites or web pages accessed using mobile devices. The application can either be written specifically for mobile devices, or the desktop version of web sites can be displayed in mobile devices, in which case the quality will not be good. The user interfaces for mobile applications are presented as a series of screens. There users are routed to various screens as they select different options. This is similar to a web application in the browsers of desktop computers where users are routed to new pages as they click on the pages. Web browser user interface control can be embedded into an application just like a button or a check box in a web form in any smartphone platform. Therefore, the entire user interface of an application can be implemented in HTML. [1, 10]

There are different browsers available on mobile devices and tablets and they all have varying degrees of support for HTML5. WebKit has become a popular layout engine for the web browsers to render web pages because of its adaptation by many mobile devices. From the start, it was designed to be lightweight and standard-compliant. In addition to the implementation of the latest World Wide Web Consortium (W3C) candidate recommendations, WebKit also implements draft-stage features. Apple's mobile and desktop Safari, Google's Android, ChromeOS, SymbianOS, BlackBerry, and webOS all use WebKit as their layout engine for their native browsers. [13, 6-7] Since

many different operating systems use WebKit, it has become very popular. When developers build an application on a browser with WebKit layout, similar user experience is offered to the user. Thus, it is possible to cover many devices with this layout engine.

Similarly, Firefox which is an open source project based on the Gecko engine, is another important player in the browser world. It is available on many operating systems; however, low end devices do not run it. The community on Firefox is its strongest point, and its compliance with the standards is very good. Furthermore, it has good support for the latest versions of HTML and CSS. It has many useful add-ons which makes development on the web easier. Additionally, Opera, another popular browser, is available for all operating systems. It is mostly recognized for its fast speed on mobile devices. [13, 7] Opera on mobile devices is known as opera mini and opera mobile. Both opera mini and opera mobile are lightened version of the browser opera and they have good support for low end devices. Dolphin is another good browser available on the Android and iOS platform.

3.1 Markup Language and Style Sheets

There are many different markup languages. HTML which is an acronym for Hyper Text Markup Language is one of them, and it is the foundation for all content appearing on the World Wide Web. HTML5 is the latest specification for HTML, and it has provided emphasis for web applications. It standardizes many features that web developers have been using for many years without documentation by a standards committee. In addition, HTML5 also provides new features which are required by modern web applications [14, 1]. It is also a cross-platform and only needs a modern web browser to run. All the major web browsers support it on a varying degree. It is backward compatible which means older browsers support it as well, but some individual features such as canvas, video, and geolocation might not be supported [14, 15].

Additionally, to describe several new tags, markup and some wonderful JavaScript APIs a specification termed as HTML5 is used. However, this specification has not yet been finalized [15, 123]. A canvas element is added as a resolution-dependent bitmap

canvas which can be used for rendering graphs or graphics. The canvas is controlled fully by the developers using their custom JavaScript file, and they have to figure out which objects are being touched by the users to fire any specific events [15, 130]. The canvas API is defined by HTML5 for drawing various shapes, defining paths, creating gradients, and applying transformations on the canvas area with JavaScript. Canvas Text API is used for text styling in canvas. However, it may not be supported by the browser even if it supports the canvas API since the text functions were added after the canvas API. [14, 16-17]

Similarly, there is a new video element defined by HTML5 for embedding video in the web pages. This removes the need to embed video with third party plug-ins such as Apple QuickTime or Adobe Flash. Old browsers that do not support HTML5 will ignore the <video> element, but they can be told to play the video with a third party video plugin. JavaScript is needed to check for the video support and video format needed by the browser to read the video. Since the video tag does not need any detection scripts to work, developers can specify multiple video files, and the browsers will select the video format they support. Video compression and decompression are done using a video codec which is defined as the language of video. Two codecs are supported in this markup language. [14, 18-20]

Furthermore, websites are allowed to store a lot of information on the computer to be retrieved later using HTML5. Although similar to cookies, the local storage of this markup language stays on the users' computers, and the website can access it with JavaScript after the page is loaded. It also allows reading the web applications offline. This works using the idea that when the users visit a website for the first time, the web server tells the web browser the resources needed by the web applications to work offline. Then the browser downloads all the resources and when the user is offline these resources are used. [14, 21] As soon as the users go online, the new changes can be uploaded to the remote web server. When the users go to a place with a mobile phone without a network such as underground, airports and public bathrooms they can use this feature and continue to browser applications offline.

In addition to this, a new feature called web workers is added to HTML5 which allows the JavaScript to run in the background. The web application is allowed to have

multiple threads such as different applications running on the computer with the web worker. While the web page responds to a user typing or clicking the background, threads can be used, for example, to access local storage or perform complex mathematical operations. [14, 23] A new feature called geolocation and a navigator are also added [15, 138]. These features give a reading of the users' current location which can then be integrated with the Google map service to show the users their current position and to show them places of attraction around them. However, for privacy, user permission is required to use this feature.

Similarly, various input types are available on the forms at web pages using HTML5. Input types include a search for search boxes, a number for spin boxes, a range for sliders, color for color pickers, a telephone number (tel) for telephone numbers, a URL for web addresses, email for email addresses, a date for calendar date pickers, a month for months, a week for weeks, time for timestamps, date and time for absolute or precise timestamps and date time-local for local dates and times. A new element called placeholder is also introduced. Using it, developers can display the placeholder text in the input field as long as the field is empty and the user has not clicked on it. Finally, it adds form autofocus at markup as well, and so the behavior will be consistent on the browsers because of the auto focus of text in web forms. [14, 27]

Figure 3 shows browser support for the HTML5 web application on different browsers on Windows and Macintosh computers.

	WIN										MAC			
														
	FIREFOX	SAFARI	IE							CHROME	OPERA	FIREFOX	SAFARI	OPERA
	3.6	4	5	6	7	8	9	10	10	11.1	4	5	11.1	
Local Storage	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	
Session Storage	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	
Post Message	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	
Offline Applications	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	
Workers	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	
Query Selector	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	
WebSQL Database	✗	✗	✓	✗	✗	✗	✗	✗	✓	✓	✗	✓	✓	
IndexedDB Database	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	
Drag and Drop	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	
Hash Change (Event)	✗	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✗	✓	
History Management	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	
WebSockets	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	
GeoLocation	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✗	✓	
Touch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	

Figure 3. HTML5 web application support by various browsers. Reprinted from [16].

As figure 3 above illustrates, the support for HTML5 is improving with each new version of the browsers. Furthermore, the browsers need to be updated regularly to make the maximum use of this new markup language. On the other hand, Internet Explorer has the lowest level of support for web applications developed using HTML5 even though Internet Explorer has the largest browser market share.

Similarly, there are different style sheets as well. CSS which stands for Cascading Style Sheets is one of them. It is used to add styles to the web applications and web documents. The appearance of websites can be controlled using CSS, and it can also help developers separate presentation from structure, thus making it easier to maintain the web sites. CSS3 is the latest version of CSS. The media queries there can be used to differentiate between various media types and display the appropriate style sheets for the media. CSS can also be used to customize the design of the application to display the page appropriately to fit the portrait or landscape view of the phones.

3.2 Frameworks

As Microsoft Windows and Apple Macintosh became popular in the 1990's, software developers needed to create applications for both the platforms. Thus, they created libraries and frameworks to abstract the differences making it easier to develop cross-platform applications [1, 9]. When the processor speed and memory can support a layer of abstraction and there is a need for market, developers find a way to create cross-platform application to ease their work. This has led to the availability of tools and frameworks for creating cross-platform application in the smartphone sector as well. To develop the web application, jQuery Mobile was selected as the framework. Similarly, PhoneGap was studied to offer native features would the application need some. It is possible to use PhoneGap along with jQuery Mobile to build a web application, and the developed web application can access the native features of smartphones as well.

jQuery Mobile

jQuery Mobile is a lightweight codebase built on jQuery and jQuery UI. It is used to create mobile web applications across a broad range of devices across all popular device platforms such as desktops, tablets, and mobiles. Developers can design and develop a web application that appears and behaves consistently across all supported smartphone and tablet platforms using the jQuery Mobile framework. To achieve these results, it has a powerful library, a standard set of layouts, a powerful theming system, user interface widgets, and a rich API. [17, 1]

Furthermore, jQuery Mobile consists of a JavaScript file, a CSS3 file and some PNG images. It uses the principle of progressive enhancement that allows the web application to work in different browsers including the ones that do not support JavaScript using graceful degradation techniques. jQuery Mobile is completely mark-up language driven and supports HTML5 which allows for fast development with less scripting. For jQuery Mobile to work, it needs the HTML5 doctype (heading at the top of the code) and links to jQuery, jQuery Mobile and jQuery Mobile style sheet. [18, 19]

Additionally, the page structure of jQuery Mobile is based on the HTML5 standard and it supports most of the tags, attributes and elements of HTML5. It also supports multipage features, which means that many pages can be included within the same page. Thus, when the developers want to give limited information on a particular page, they can use this feature and add multiple pages in one page. When a multipage template is used, the first page is the default page. [18, 36]

Similarly, jQuery Mobile allows internet and external linking of pages. Internal pages have the same domain as the currently browsing pages while external pages have a different domain. Internal linking uses Ajax-powered navigation that allows animated page transitions while external linked pages are loaded in the normal way. Ajax is used to asynchronously request particular information instead of loading the whole page from the server which results in a smooth and pleasant user experience. [18, 35-37] The transition from a popup page to a normal page is also quite smooth using the jQuery Mobile framework. Appendix 1 shows the Ajax navigation in an application using this framework.

Touch events with jQuery Mobile are triggered every time the users touch the page. This feature works on desktop computers, tablet computers and smartphones alike. Touch events on desktops are triggered with tap and swipe using the mouse. [18, 55] Five themes are available there by default and more can be added by editing the CSS file. A select menu similar to that in the iOS and Android is also available using the jQuery Mobile library. The menu options appear as a drop down box with the select menu. The elements needed in form are built by the jQuery Mobile framework on top of native HTML5 elements. JavaScript can be used for form validation on the client side. Appendix 3 shows the various themes and the available select menu using the jQuery Mobile framework.

Finally, the user interface elements of jQuery Mobile are created by writing semantic markup and then applying a data attribute to the element. The elements are enhanced upon initialization with the available framework. Two standard jQuery development patterns, the plug-in and the widget are supported. Developers can add custom methods and encapsulate an application, page initialization, functions and data managers using the jQuery plugin. The jQuery widget pattern can be used if

developers need a more powerful plugin. Garbage collection is enabled, which makes it possible to avoid memory leaks in browsers. jQuery Mobile favors compatibility across different devices, and many functions are available. [17, 25] Figure 4 below shows the level of support for browsers and PhoneGap targeted by jQuery Mobile at the start of the project.

MOBILE GRADED BROWSER SUPPORT														
Platform	Version	Native	Opera Mobile				Opera Mini		Fennec		Ozone	Netfront	Phonegap	
			8.5	8.65	9.5	10.0	4.0	5.0	1.0	1.1	0.9	4.0	0.9	
iOS	v2.2.1	A											A	
	v3.1.3, v3.2	A						A					A	
	v4.0	A						A					A	
Symbian S60	v3.1, v3.2	C	C	C		B	C	B			C	C		
	v5.0	A	C	C		A	C	A					A	
Symbian UIQ	v3.0, v3.1			C							C			
	v3.2				C						C			
Symbian Platform	3.0	A												
BlackBerry OS	v4.5	C					C	C						
	v4.6, v4.7	C					C	B					C	
	v5.0	A					C	A					A	
	v6.0	A						A					A	
Android	v1.5, v1.6	A											A	
	v2.1	A											A	
	v2.2	A				A		C		A			A	
Windows Mobile	v6.1	C	C	C	C	B	C	B				C		
	v6.5.1	C	C	C	C	A	C	A						
	v7.0	C				A	C	A						
webOS	1.4.1	A											A	
bada	1.0	A												
Maemo	5.0	B				B			C	B				
MeeGo	1.1	A				A				A				

Figure 4. jQuery Mobile support for browsers and PhoneGap at different platforms. Reprinted from Giulio [18, 18].

As illustrated in figure 2, jQuery Mobile can be used to develop web applications for many different platforms. The A grade in the figure dictates high quality support, the B grade dictates medium quality, and the C grade dictates low quality. Bug fixes are applied to A and B level browsers but not to C level browsers. The C level browsers are not capable of utilizing media queries and they will use plain HTML. [18, 19]

PhoneGap

PhoneGap is an open source development framework that allows the developers to use a markup language (HTML), a style sheet (CSS) and a scripting language (JavaScript) to develop applications similar to native applications for various platforms such as iOS, Android, BlackBerry, Palm, Symbian and webOS [15, 1]. PhoneGap is used to wrap a mobile application in a stand-alone application. It provides access to native APIs including a camera, media, geolocation, contacts, an accelerometer, a compass and other sensors. This allows the developer to create applications for a wide variety of clients.

Additionally, PhoneGap can be termed as a set of project templates for different mobile operating systems where the developers can develop applications in a consistent fashion without knowing too much about the SDK of all the various operating systems. If a vendor releases SDK only for a particular operating system, the developer has to use that operating system to develop applications with PhoneGap. This is normally the case with iOS. The developers need to have Macintosh computers to develop applications for iOS and Windows PCs to develop BlackBerry applications. The simulator for Blackberry only runs on Windows machines. The developers can use any of the following platforms: Windows, Mac OS X, or Linux. On these platforms it is possible to develop applications for Android, webOS, and Symbian.

Similarly, an application developed with PhoneGap can be tested with any browsers, but browsers with the latest updates are recommended because they have better support for the new version of the markup language and style sheets. For maximum benefit, applications should be tested with browsers using WebKit since all the major OS vendors now use WebKit, and PhoneGap has maximum support for it. PhoneGap uses HTML5 that allows the applications built on the mobile web to match and even exceed the quality of native applications. [15, 123] Using the native APIs, the capabilities of the web application can be increased tremendously, which lets the developers add native features to the applications, and the performance for the tablets and smartphones can be increased as well.

3.3 Web Services

A web service is a network accessible interface developed using standard internet technologies to exchange data between applications. Any application accessible over a network using any protocols is a web service. It acts as an abstraction layer between the application code and the application users. This enables any programming language on any operating system which supports the web services to access the application's functionality allowing for cross-platform interoperability. Web services enable the developer to develop software systems from distributed components. For any web application to be a web service, message transfer should be allowed between two clients using standard internet protocols. XML is often used for communications between different systems since XML is in text format, and almost all systems can understand it. [19, 1-2]

It takes a lot of time and resources for companies to develop all the application components they need in-house. Furthermore, after the development, many hours are spent maintaining the quality assurance. This includes testing, debugging and fixing the problems. By using the web services, the developers can use available, proven, and tested applications as part of their own application if they want to increase the number of features of their applications [20, 16]. For example, if the developers want to add a feature to share photos, they can use Flickr or Picasa API as part of their application. This saves a lot of resources for the company. Thus, developers become more productive when they use web services as they can devote more time to develop features for the application.

Nonetheless, there are different types of web service frameworks. Windows Communication Foundation (WCF) is one of those. WCF is an SDK developed by Microsoft for building and deploying services on Windows. It allows the developers to integrate multiple technologies at the same time allowing interoperability between services, too. The runtime environment can be provided to the services using WCF which enables the developers to expose Common Language Runtime (CLR) types as services and other services as CLR types [21, 1].

WCF is part of .NET 4.0 and can run on many operating systems: Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008 and Windows 7 or later [21, 2; 22, 1]. WCF usually has SOAP messages, and the messages can communicate over a wide variety of protocols not limited to HTTP unlike web services. The client uses a proxy to forward calls to the service with WCF. Hence, this allows it to use the same operations as the web services and also proxy management methods [21, 3].

WCF is a truly service oriented interoperable platform. Figure 5 shows distinct WCF services consumed by clients using .NET and Java programming languages. The WCF service developed for .NET application can be consumed by the Java application and vice versa.

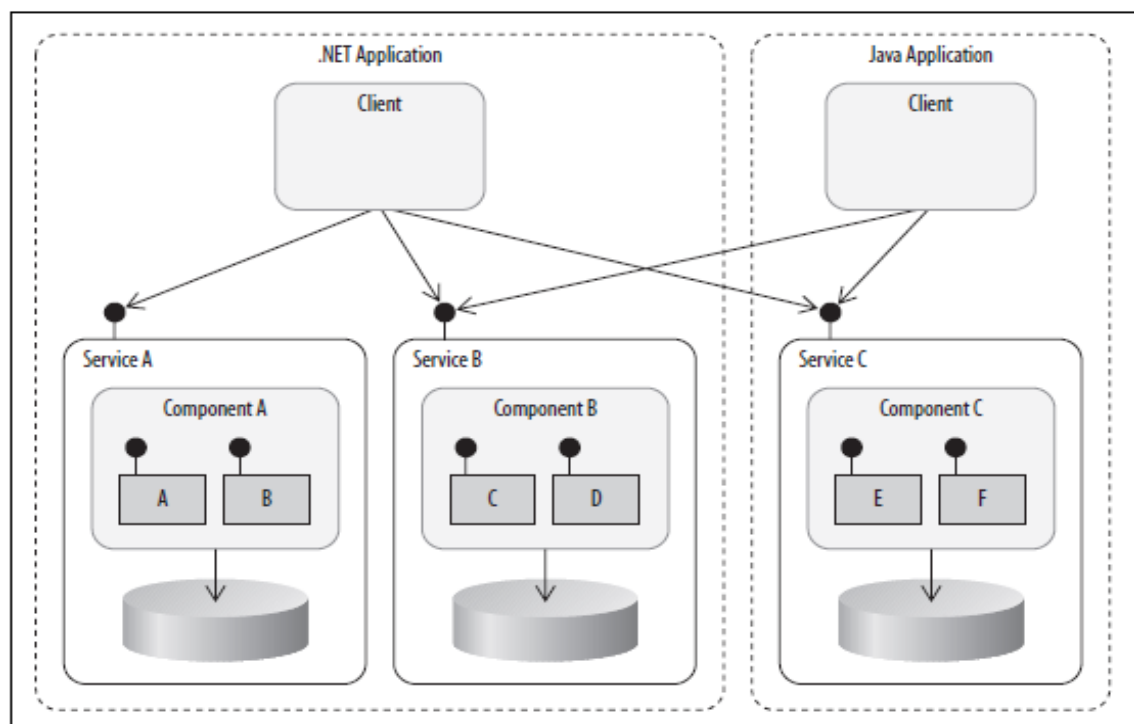


Figure 5. Cross machine communication using WCF. Reprinted from Bustamante [22, 5].

As figure 5 illustrates, developers can use the same WCF service for different clients. The clients can interact with services on local intranet or across the internet. Clients can be a Windows form, a Silverlight class, an ASP.NET page, a Java application or another service. This is particularly important because there are many different technologies available, and a single web application might be using various languages. Also, today the enterprise system is developed with different functionalities across processes and various machines to serve clients using distinct languages [22, 4].

Similarly, secure interaction between a client and a service can be maintained using several processes. Initially, the service authenticates the callers while connecting to determine that the callers are who they claim to be. Then the callers are authorized before executing sensitive operations. After that, the message en route from the client to the service needs to be authenticated to secure the service and clients for any distributed systems. When the message arrives securely in the service after authentication and authorization, the service has different options to execute the operation depending upon different service frameworks used. For maximum security, all these security techniques must be followed. [21, 525]

Authentication relates to identifying the message sender and involves identifying both the sender and receiver to prevent a man-in-the-middle attack. After authentication, authorization determines what the callers are permitted to execute. Integrity defines that the messages between the sender and receiver are digitally signed to verify that the message contents have not been altered. Finally, confidentiality determines when the message needs to be encrypted to verify that they cannot be openly viewed. Integrity and confidentiality are collectively termed as transfer security as well. [26, 408] The mutual authentication mechanism for transfer security is also needed to detect and eliminate replay attacks and Denial of Service (DOS) attacks [21, 527].

Furthermore, different security policies can be created and runtime behaviors can be set to control security features with WCF services. WCF services support a variety of mutual authentication mechanisms using token formats such as Windows authentication, username and password, certificates, custom mechanisms and issued tokens when applicable. Windows roles, ASP.NET roles such as SQL provider or custom authorization policies can be used for authorization with WCF. [22, 408]

Finally, symmetric or asymmetric keys can be used in WCF for message protection (integrity and confidentiality). Using it, developers can use the existing security infrastructure to secure communications behind the firewall as well as enable the latest in Simple Object Access Protocol (SOAP) message security standards for secure and interoperable messaging with other applications, platforms and business partners. Many security settings in WCF are exposed as part of a service security policy which can then be easily used by the clients. [22, 409; 19, 526-527]

4 Method and Materials

Unlike application development for native platforms, web application development does not require a specific integrated development environment (IDE) and an SDK which saves developers' time. For native application development, developers need to download different IDEs and SDKs for each platform, which is very time consuming. However, for web application development, developers can use any text editor tools, and they can test the application with the web browsers. Flot, which is a Javascript plotting library for jQuery, was used to generate all the graphs for the web application.

4.1 Development Tools

Xampp is a software platform which enables the user to run the website on their computer without uploading the website on the server. It works without any internet connections. Even if the client browser and server reside on the same physical computer, the server and PHP interpreter must still be installed and configured to run the web application [23,172]. For this purpose, Xampp has a precache apache server to run the website and a MYSQL database to store the data.

Furthermore, the various letters in XAMPP have different meanings. X stands for cross-platform, A for Apache HTTP Server, M for MySQL, and P for PHP and Perl. Developers can develop web applications on their local machine and deploy them in XAMPP without uploading them to the web server. This functions like developing in the real web server. Thus, this saves the developer a lot of time and also reduces website failure should the website be tested in a real environment.

Similarly, Dreamweaver is a web development application tool developed by Adobe and Dreamweaver CS 5.5 is the latest release version. It allows the users to preview websites natively in a preview pane with a live view option. The design view option in Dreamweaver also makes it easier for the developer to make small changes in the code and see immediate change. Dreamweaver CS 5.5 integrates the jQuery Mobile framework, which is a cross-platform, and allows rapid web application development. It is also possible to develop native mobile applications for iOS and Android with PhoneGap.

The ability to view pages with different screen resolutions without leaving the document window is one of the key features of Dreamweaver CS 5.5. Furthermore, Dreamweaver CS5.5 has integrated support for PhoneGap, and the developers can choose to develop the application using PhoneGap and jQuery Mobile just by selecting the readymade templates available there. After that, the developers can start building the application with various tools. In this project, the web application was also deployed at the apache server in Metropolia University of Applied Sciences. The application was then tested with real smartphones and an iPad. In addition to this, some virtual emulators were also used to test the application.

4.2 Overall Design

The project was developed using the MVC model design plan. The MVC approach improves development by breaking the application into three distinct components: the model, the view and the controller. The business logic is separated from the presentation, and the controller is placed between them for them to communicate. With MVC, each component can be developed and maintained in isolation. Using MVC, the business logic can be reused in some other projects as well, and thus developers' time is saved when rewriting the same code. The code will also be much easier to read and understand. Web designers can take care of the presentation while the programmers can develop the business logic. The different parts of the applications can also be maintained and tested independently.

Similarly, the model holds the real business logic and the state. It defines the process an application is intended to represent. In addition to this, the model communicates with the database if needed and sends the appropriate information back to the controller. The controller determines the response of the application by communicating with both the model and view. It uses the user input from the request, which it gets from the view, and sends it to the appropriate model. It is also used to route all requests to the appropriate controller and to return the response. Finally, the view is used to format the data returned by the model and to present it to the users. The view receives the state of the model from the controller. [24, 604]

When the users request for any page or click any link, the request directs the appropriate controllers through the routing system (index.php in this application). Any values or parameters sent by the users are sent along with the request to the controller. The controller then uses the appropriate model with any values provided by the user if applicable. The model will connect to the database with the stored credential on the web server if necessary. When the model connects to the database, the necessary Structure Query Language (SQL) commands are executed and values from the database are passed to the model. The model then sends all the data to the controller which then displays the appropriate view page along with the output data. The controller might also use the helper class to display the view directly if no processes need to be done. Figure 6 below displays the MVC scheme used for the development of the application.

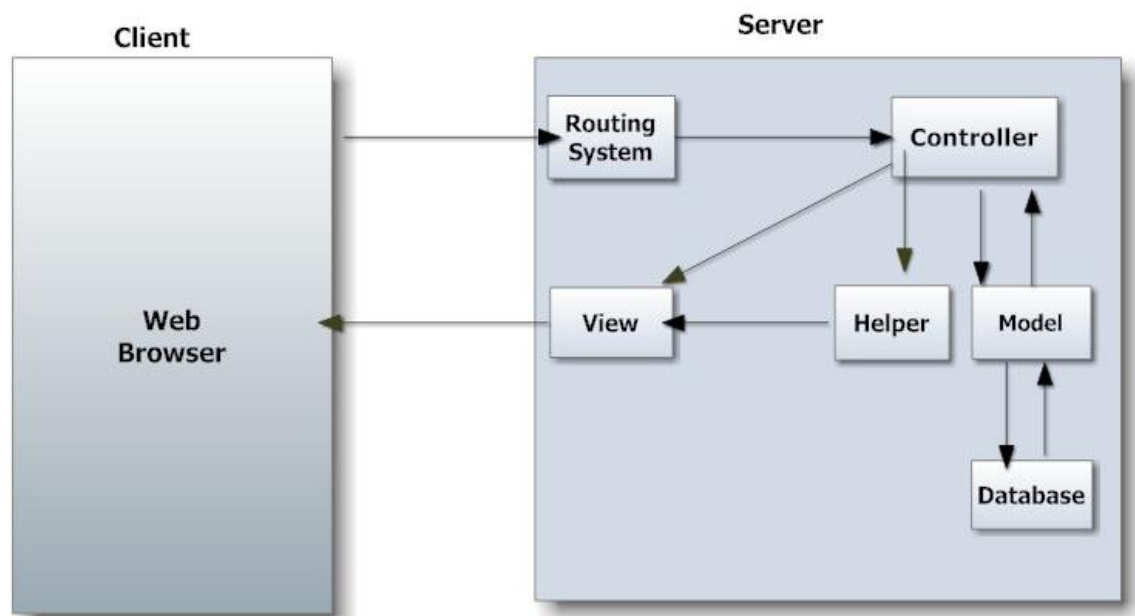


Figure 6. The MVC model used for the web application development.

Initially the client uses the web browser to open the web application. Then as seen in figure 6, after the client clicks a link, the client request is routed to controller which then uses the appropriate model to display the correct view. The controller might also display the appropriate view using the helper class if no processing is needed or the web application has to display a static page situated on the web server. After that, the same action continues. The JavaScript files for jQuery and jQuery Mobile along with the jQuery Mobile style sheets are used directly from the official website of jQuery Mobile by adding them as a reference.

Furthermore, the web application was divided into five folder structures: config, css, assets, lib and image. In addition to that, there was an index file for routing the user requests to the appropriate controller for displaying the views. Database connection information, helper files and information from all the controllers and model classes were stored in the config directory. Basically config was used to store all the configuration files. A personal JavaScript file to check form validation and a date picker for choosing different dates were stored in the folder lib.

Similarly, four custom style sheets were stored in the folder css. One style sheet was used to differentiate between different devices for displaying the appropriate style sheet from the list of style sheets stored there. The remaining style sheets were used to display the web application appropriately for a various set of devices. Small-screen and medium-screen smartphones had one style sheet, large-screen smartphones had another style sheet, and finally the tablet had the remaining style sheet. It is also possible to add more style sheets to cover more devices with different screen sizes.

Additionally, the model, view and controller were all placed under their respective headings in the folder assets. Finally, the images in the application were stored in the folder called image. There were distinct images for different style sheets: small, medium and big. Flot, which is a JavaScript plotting library for jQuery, was used to generate all the graphs for the web application. In general, different graphs can be created using Flot, and Flot focuses on simple usage, attractive looks and interactive features such as zooming and mouse tracking. With Flot, the developers can develop unique type of graphs such as bar diagram, lines, points, filled areas, and their combination in same or different plot.

In this project, CSS media queries were used to distinguish between small-screen and large-screen smartphones and tablets to display the appropriate layout. The CSS media queries were particularly important to display the graphs at various devices with appropriate scales. It is also possible to expand the content of the application at landscape and portrait view by using another style sheet for a screen wider than certain length. When the device is held with landscape view, the width is greater than the height, and thus the graph width is increased accordingly. There is more data in this view and the graph covers less area.

Figure 7 shows the graph drawn using Flot. The graph uses a bar diagram and lines to represent temperature and energy consumption at different time periods.

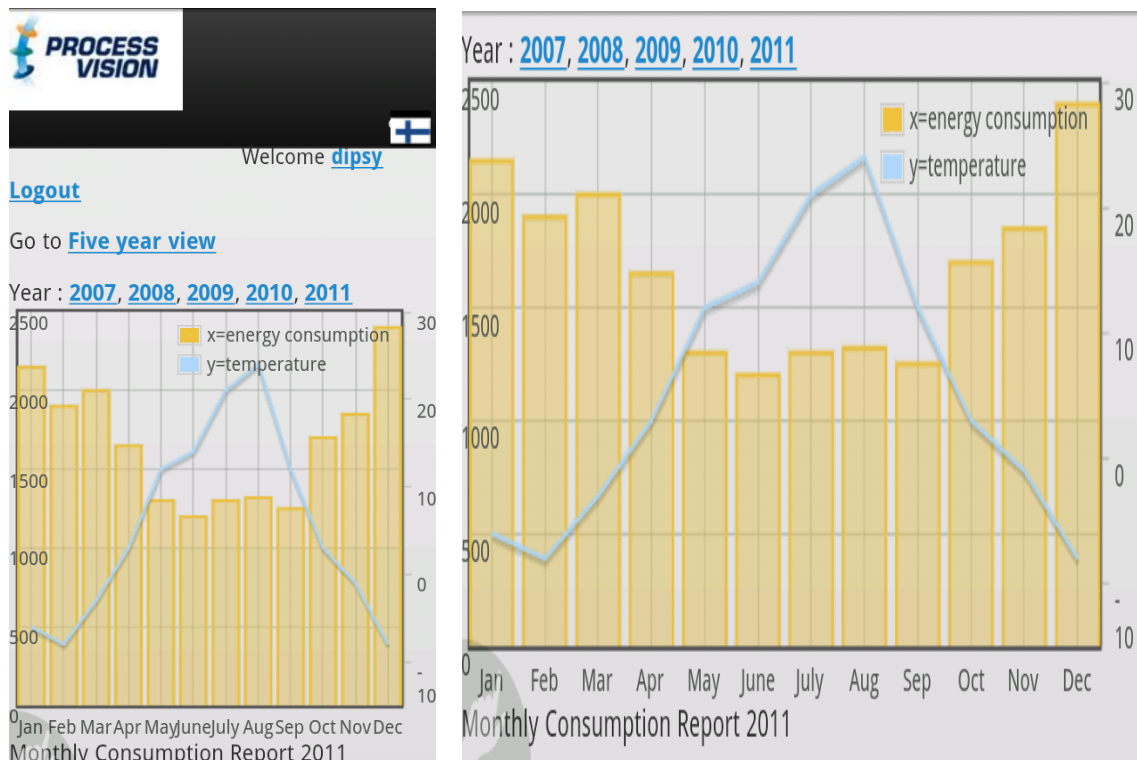


Figure 7. Portrait and landscape view of the same graph in a smartphone using Flot API.

As can be seen in figure 7, the content expands according to the size of the device. When the smartphone is held with portrait view, the height is greater than the width, and thus the graph is smaller. More content is shown in this view, and the users do not need to scroll much to see all the content. The graph is plotted against different months at a particular year (2011 in this case). When the device is held with landscape view, the width is much greater than the height, and thus the graph width is increased accordingly. There is more data in this view and the graph covers less area. The users can select a different year, and the graph will change accordingly. Alternatively, the users can select a five-year-view, and the graph will show the plot value for five distinct years. This can be useful if the energy and temperature consumption for various years are compared.

5 Application Description

This application allows the user to get information about their energy consumption and temperature for the selected time intervals. The users can also track their contract with different service providers and check their subscribed services. To use the application for maximum benefit, the users need to have a contract with the service provider which means the organization running this website. If the users have a contract with the company, they can use the contract details such as the customer code, contract code and metering code from the invoice to create a new account in the database. Once the users have created an account, they can log in and see their energy consumption for various months and years and compare it.

Similarly, users can see their invoice breakdown, contract details, and energy price at different hours. After that, the users can use energy when it is the cheapest. Thus, when more customers use energy at varying hours, the energy consumption will also be uniform throughout the day, and there will be balance in the energy consumption with less peak hours. Alternatively, the web application helps the users to keep track of their bill payments and gives them good advice of how to use energy efficiently. The application is aimed to help the user see their energy consumption and help them use it carefully.

5.1 Architecture

In this application, there is one database, and it has seven different tables. The first one is an invoice table which records all the information about distinct customers. This table has four unique fields, and the field `customer_code` is the primary key in this table. Once the account has been created for a particular customer, the field `account_created` will store value true, and no more accounts can be created for that user. The second table is people, and it has three different fields. The field `username` is the primary key in this table while the field `customer_code` is the foreign key. Once an account is created for a particular user, they get a unique username for the account with that particular `customer_code`. The third table is personal which has the field `email` as the primary key and the `username` as the foreign key. There are ten fields in this table. This table is used to store all the information about the user.

Similarly, the fourth table `customer_contract` has eight distinct fields where the field `customer_contract_code` acts as the primary key while the fields `customer_code` and `contract_code_type` act as foreign keys. This table is used to store all the contract details of the customer. The fifth table is the table `contract`, and it has the field `contract_code_type` as the primary key. This table is used to store information about various contract types and the price for different times. After that, the sixth table `consumption` displays the graphs for energy consumption. There are seven fields in this table. The field `con_id` is the primary key, and `customer_code` is the foreign key in this table.

Using all the customer details, graphs are drawn monthly or yearly with energy and temperature as the graph elements. Finally the table `consumption_realtime` has three fields. It has the field `con_realtime_id` as the primary key, and `customer_code` as the foreign key. This table is used to generate a graph for displaying real time energy utilization with less time intervals. At present, graphs can be drawn for various days. However, the aim is to further expand this feature and draw graphs for different hours. Figure 8 below shows the relation between the different tables.

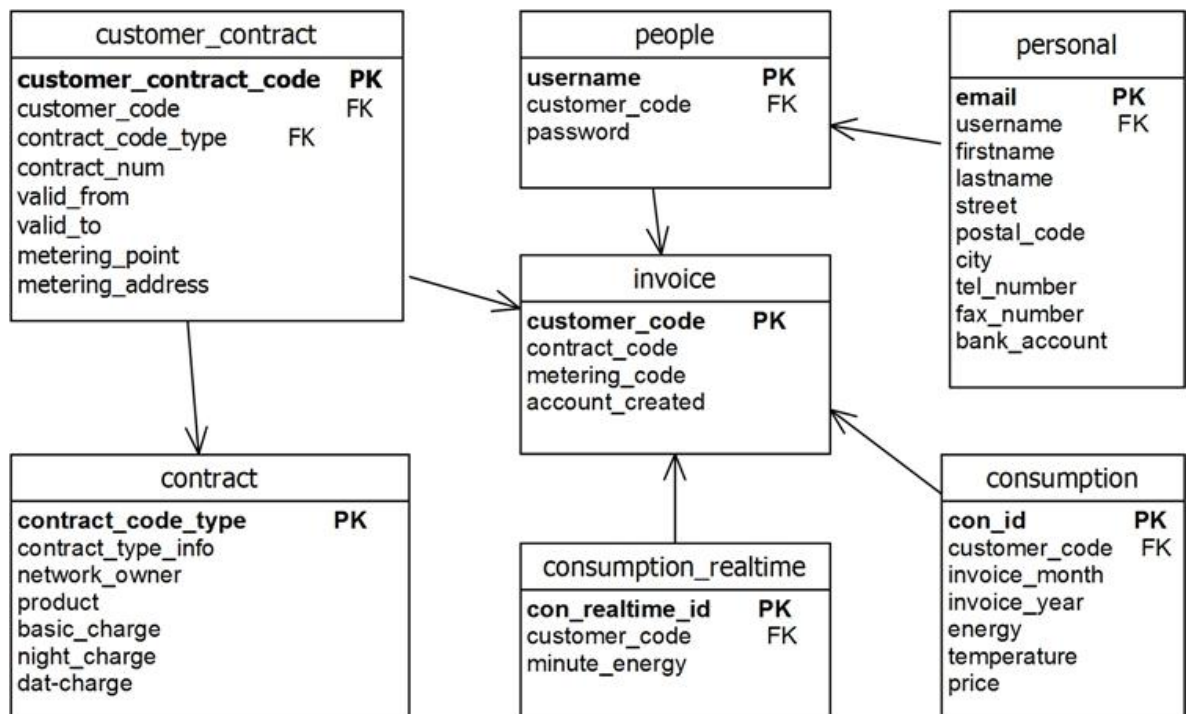


Figure 8. Table relationship in the database.

Figure 8 is used to depict the relation between the different tables in the database. As can be seen in the figure, all the tables excluding the table contract are related to the table invoice. They have the primary key from the table invoice as the foreign key to distinguish between different customers. The table contract is used to provide additional information to the table customer_contract. The data there includes the information about the different network owners and the price at different hours. All the forms used in the application use JavaScript form validation for client side validation.

5.2 User Interface

User interfaces allow the users to communicate with a program using commands and menus. They can make a difference between the success and failure of an application which is determined by the application downloads. They also help to determine how easily the users can use the application in a way the developer wants. Users want all of the applications they use to look and feel the same. They do not appreciate the sudden change of user interface of an application unless the software or the device is a trend setter such as the iPhone.

Similarly, the web application offers similar experience as native application across all the platforms through its use of menus and buttons which are readily available. In addition to this, information is present as collapsible content on various pages which helps to save the space on mobile devices and tablets. There are five distinct themes available for the users to choose from and each theme has a unique color. When the users select template, the selected theme option is sent to different pages using the GET request, and the appropriate theme is displayed.

Furthermore, all the forms in the application are validated using JavaScript validation. In this form, the forms are validated to check if they are empty, if they contain letters where only numbers are allowed, if length of the field members and password criteria are correct, and finally if the email format is correct. Some pages in the web application make use of a popup page for unique user experience. Appendix 2 shows the JavaScript validation and the popup page disappearing after the user has pressed the back button in the web application.

Figure 9 displays the user interface of the web application. The first picture shows the user interface the users get when they open the web application, while the second picture shows the user interface when the users log in the application.

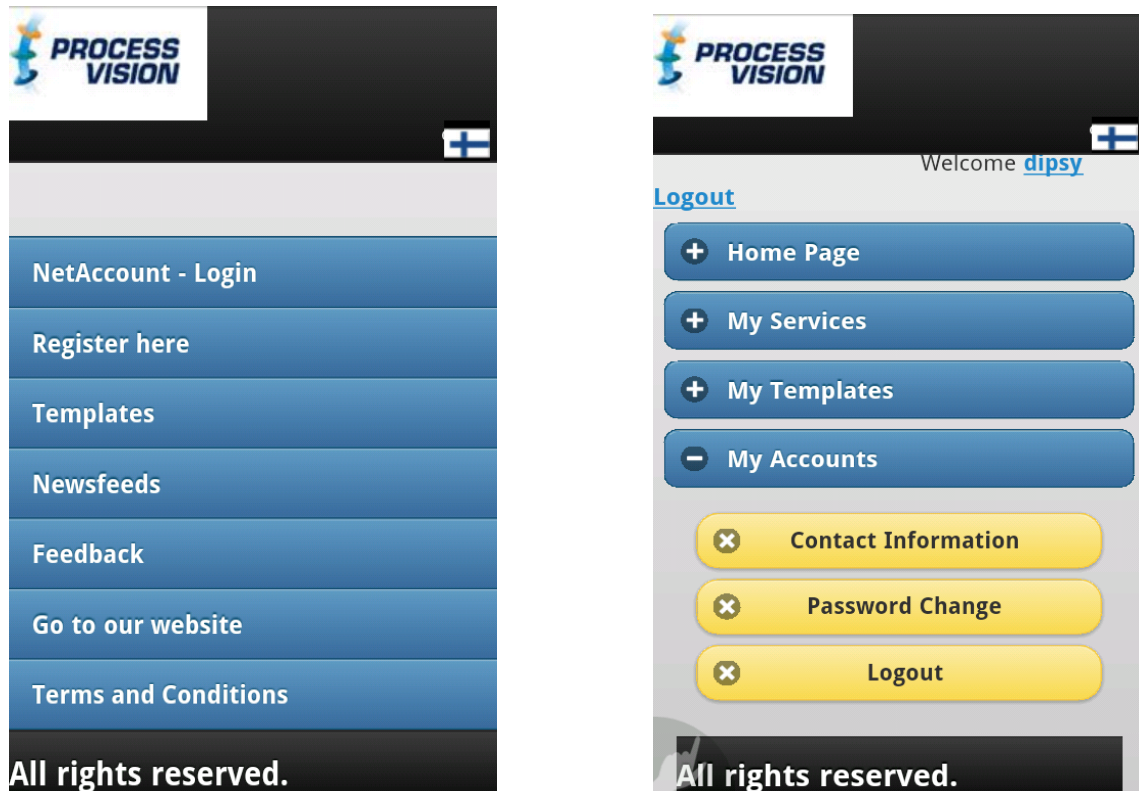


Figure 9. User interface for the front page and the success page.

The logo of the service provider company is situated at the top left corner of the screen, and the language option is at the top right corner of the screen just below the logo of the company as seen in figure 9. The different language option is displayed using the flags of the appropriate country. When the user is using English to view the web page, the Finnish flag is shown on the screen and vice versa. Sub-menus are placed as collapsible content under parent menu options to save space. This is done to show all the menus without the need to scroll.

After the users log in the web application, their names are displayed just below the flag, and the option to logout is situated below the flag on the left side. The users can use their fingers to select menus in mobiles and tablets. Additionally, on tablets and on desktops, it is also possible for the users to use a mouse to do the tasks accessible using touch.

Similarly, the web application has various views, controllers and models for different application pages. When the users first log into the web application, they can choose between seven different menu options. Each menu option opens a distinct page. With the first option, the users can login to the application. If they enter the correct credentials, they can enter the member page. Otherwise, they might need to re-enter their credentials or reset their password if they forget their access information. They can do this by selecting the menu option "reset pass" by username or by email inside the "password forgot" menu.

Additionally, the reset pages appear as a popup page. With the template option, the users can select between five distinct themes. According to the users' choice, the page changes to different colors. The third menu option is newsfeeds, and when this menu is selected, the users can check news. The fourth menu option is register with which the users can register themselves to log into the web application. The users need to enter valid customer information in the first register screen to verify themselves as authentic customers. Then they have to enter information such as username and password to log in the website. Finally, in the third register screen, the users can enter their email address which is useful should the users forget their access information.

After the users register themselves in the application, the success page is displayed, stating that the login was successful. The fifth menu option available in the application shows feedback. With this option, the users can send any information to the company. With the sixth option, the users can visit the website of the service provider, and finally with the last menu option, the users can see the terms and conditions to use the web application.

In the project, the forms allowing only numbers as input were created where necessary with HTML5, and placeholders were placed to show the users what needed to be placed at various fields of the form. In general, all the forms used in the application use JavaScript form validation for client side validation. It is also possible to deny the form submission with HTML5 if any of the field is empty; however, this feature was not supported by all the browsers. Thus, JavaScript was used to check this as well.

Figure 10 below illustrates the web site structure when the user opens the web application. This is the first screen that the users see in their screen.

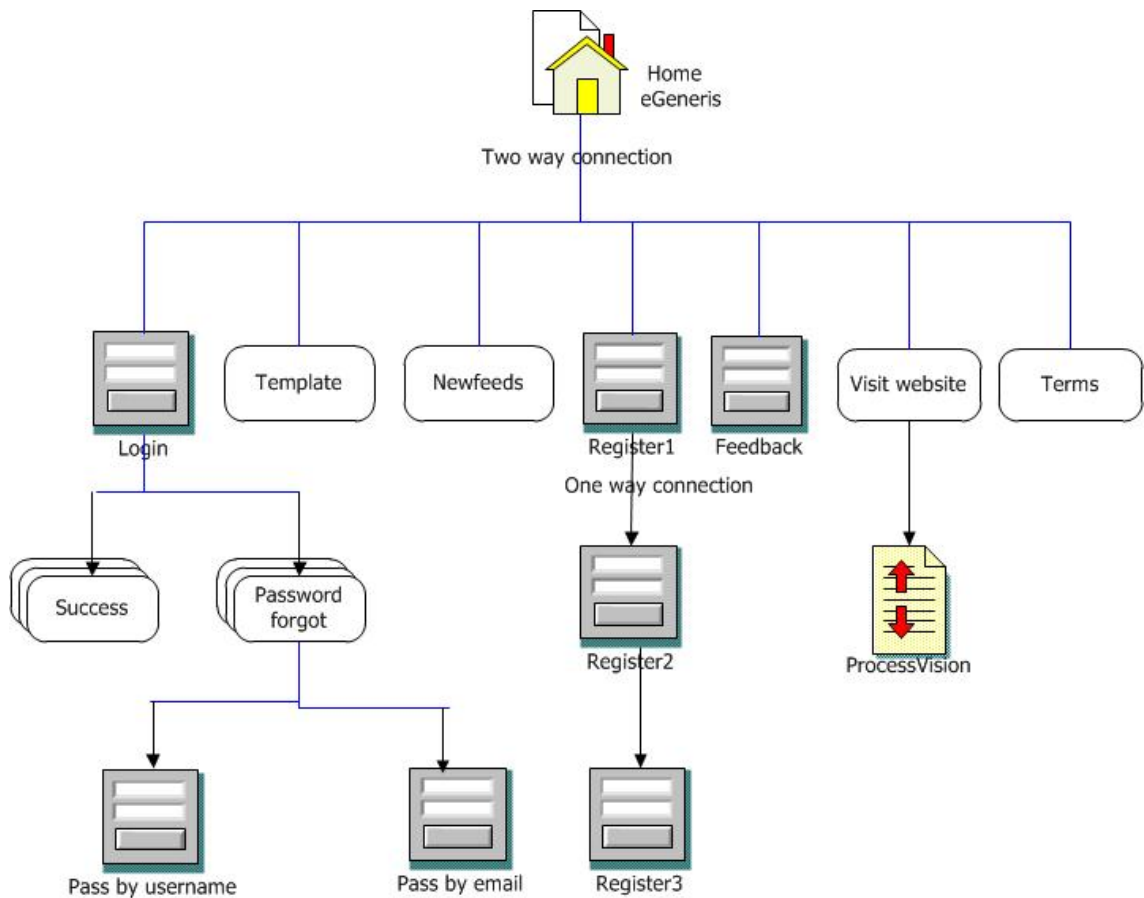


Figure 10. Web site structure when the user first opens up the application.

As can be seen in figure 10, eGeneris is the home page of the application which the users get after opening the web application before they login. The users can be on any page and still navigate to the home page by clicking the logo of the company. The logo of the company acts as a navigation button to the home page in this regard. Many pages can be opened by selecting different menus in the home page. When the users click on the visit web site menu, an external web site link called Process Vision opens up.

Additionally, the two way connection in the figure shows that when the users navigate to the child page, they can return to the parent page by clicking the logo of the company in this application. The one-way connection signifies that after that the users have navigated to the next page, they cannot navigate directly to the older page unless they press the back button.

Similarly, the success page acts as the home page after the users have logged into the website, and whenever the users press the logo of the company, they are directed to the success page. The user can visit the front page of the web application by selecting the menu option "home" and then choosing the option "front page". After the users navigate to the front page, this page acts as the home page, and if they login once more, the success page acts as the home page.

After the users' login to the web application, they are routed to the success page. There are four different menu options with collapsible content on the success page. The menu options can be collapsed further to select sub menus. Each menu option has three to four sub menus under them. The sub menus further open up a new page when selected.

Using the first menu option on the success page, the users can go to the front page, see a video, read news and tips to save energy, and visit the company website through an external link. News and tips have different content with a heading as collapsible content. The second menu option is used to provide services for the users. Services include checking out energy consumption, forecasted energy consumption, report of their energy consumption at various temperatures, and different contract/s available to the users.

Energy consumption can be observed for various days in case of real time consumption, and for different months and year for normal consumption. The forecasted energy consumption predicts the energy consumption by the users for the next month or year. The energy consumption has been forecasted by taking the mean of energy consumption at selected time intervals. They can also look at the report of their energy consumption at distinct times of the year, check the temperatures at the selected time interval, and compare it with another time interval. Finally, with the contract option the users can check their contract and energy price at different times of the day. They can also see all their subscribed services.

Furthermore, the users can select unique themes from the third menu on the success page. There are five templates available for the users to choose from. The themes menu present in the success menu option is similar to the template menu present on

the front page. Finally, the fourth option menu is my account. With this menu, the user can see their contact information and make changes if needed, change their passwords or logout. Figure 11 below depicts the menu option available to the user after they login to the web application.

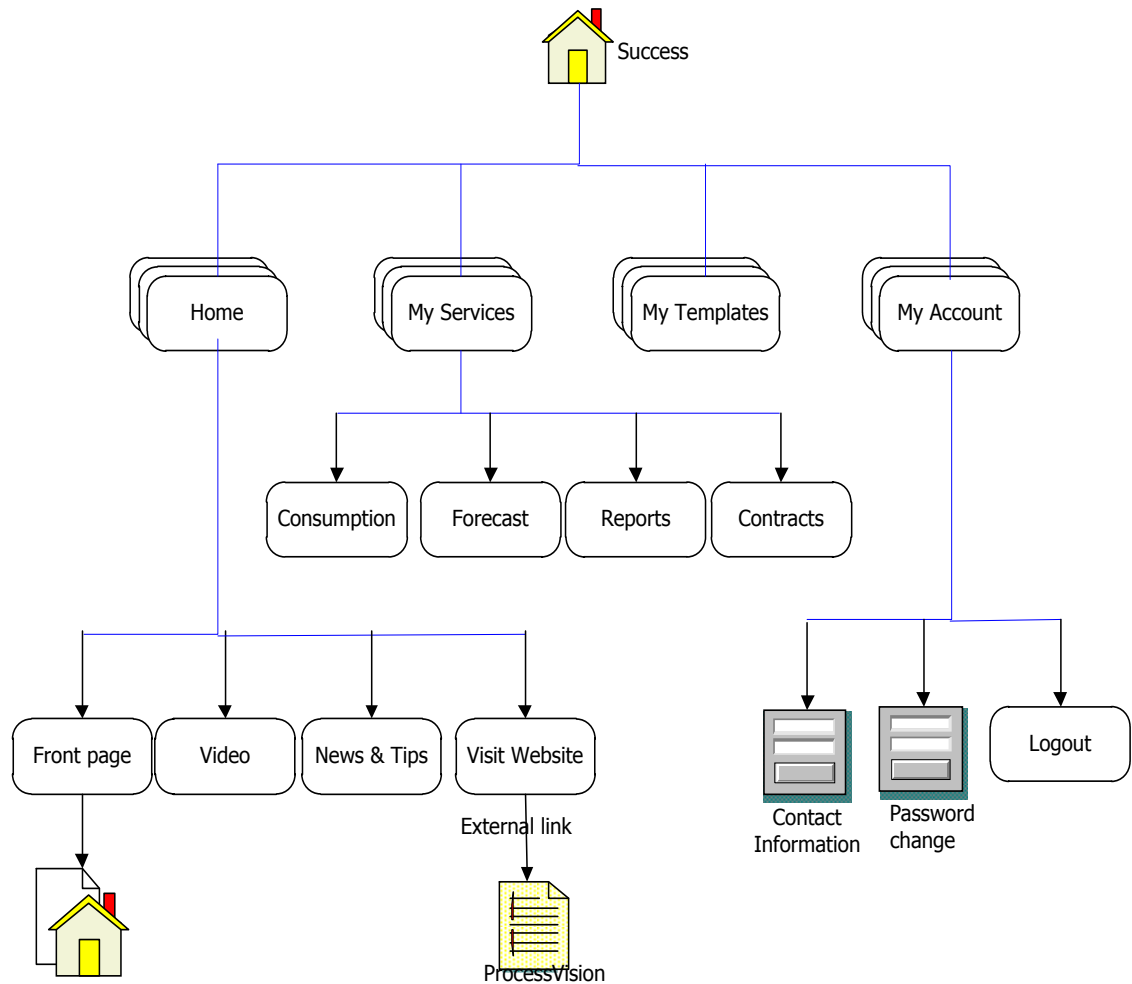


Figure 11. Website structure after the user logs in the success page.

As can be seen in figure 11, the website structure on the success page also has a two-way connection as in figure 10. The only major difference in the menu mechanism between figure 10 and figure 11 is that in figure 10 each menu option opens a new page; however, in figure 11 the sub menus are available as collapsible content. With collapsible content, the users can view the sub menus on the same page as the content headings. Hence, it saves the buffering time and gives smooth page transition as well.

5.3 Web Service Recommendation

For secure data connection, WCF services can be used. WCF checks for certificates from the web server and does not give direct access to database which helps to make the application secure. If the application is developed using WCF services, the web server and database can be maintained separately by different groups or companies, and they do not need to give full access rights to one another. Thus, it is much easier to maintain the application. It can also be used if there are multiple applications with different languages trying to access the web service. Figure 12 below illustrates how the web application can be made secure using WCF services.

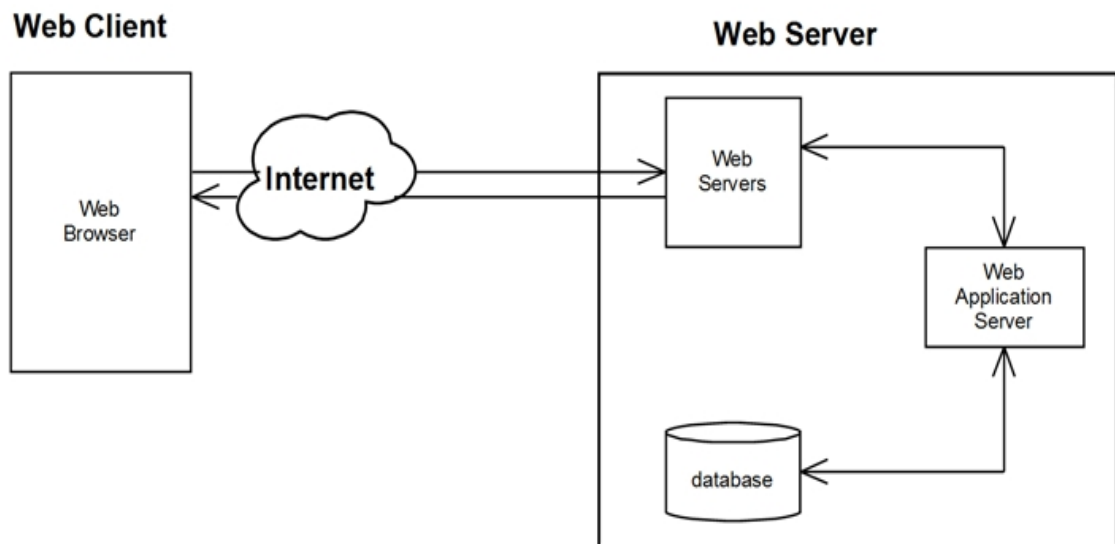


Figure 12. Secure application connection with web services.

As illustrated in figure 12, the web client requests for data with the web server which in turn connects with the WCF services with the available certificate preventing unauthorized web servers' access to the WCF services. If the certificate is authenticated, the WCF services allow the web server access to the database. The web server in this case does not have direct access to the database. It does not matter if the web and the database are on the same web server or if they are situated on two different web servers, the process is the same. The web server is authenticated by the WCF services and allowed access to the database, and thus increasing the level of security.

Similarly, various companies with different web servers can access the same database using WCF services. This helps the company to lower costs. For example, company X sells an application to other companies, company Y and company Z which then use it for Customer Service Management (CMS). Company X can use the same database for both company Y and company Z. Hence, the cost is reduced for all the companies, and privacy is also maintained since company Y and company Z cannot access each other's database.

6 Results

After the mobile web application was created, it was tested with devices from various companies using different operating systems. Then, the application was also tested using various browsers from the same device to check the browser capability across different platforms. This was done to observe the performance of the application and ensure that the application met the cross-platform requirements. The web application was tested with wireless internet through the network operators and also over Wi-Fi for checking at various internet speeds. Table 2 below shows the performance of mobile application with various devices running different operating systems.

Table 2. Performance at various browsers by devices using different operating systems.

Operating	Brand	Model	Performance			
System			Native	Opera Mini	Firefox	Dolphin
Android 2.2	HTC	Wildfire	A	B-		A+
Android 2.2	HTC	Desire	A	B-	B	A+
Android 2.3	HTC	Desire HD	A	B	B+	A+
Android 2.3	Samsung	Galaxy S2	A	B	B+	A+
iOS 4.0	Apple	iPhone 3GS	A	B		A
iOS 4.0, 5.0	Apple	iPhone 4	A+	B-		A
BlackBerry 7.0	RIM	Torch9810 virtual	A+			
Windows Phone7	Windows	Virtual	A			
WebOS	HP	Virtual	A			
Maemo 5	Nokia	N900	A	B	B	
Symbian	Nokia	XpressMusic5810	C	B+		

As can be seen in table 2, the web application performed well in all the devices. Here symbol A+ shows that the application performed quite well, supported all the features and looked very nice. A showed that the application performed quite well and supported all the features but the display could have been made improved a little. B+ means that the application performed well, supported all the features but needs to improve speed and the display of the application.

Similarly, B shows that the application performed well but needs to improve the speed and the display of the application. B- means that the application worked fine but the application did not support all the features and the display needed improvement. Finally, C means that the application did not perform quite well with work needed to support all the features, and improvement was needed for both speed and display. Since physical devices for all the operating systems were not available, virtual devices were used to test some operating systems such as BlackBerry OS version 7, Windows Phone 7, and webOS from HP.

Surprisingly, it was noted that the performance of the web application was better using dolphin HD than using the native browser on the Android platform. Opera mobile/mini, which is known for its speed, was indeed found out to be fast, but the quality of the application was not good compared to other browsers. It was good in smartphones with mid to low range specimens, but when it was tested on high end devices, page rendering was not too impressive, and it did not support all the features such as displaying the graph made with Flot. Opera Mobile in iPhone 5.0 and the native browser in Symbian OS had no support for the Flot as well. In addition to that, the performance on that device was not quite up to par with other operating systems such as Android and iPhone.

There is the possibility of error in this study since there are a lot of devices using the same operating system, and only a handful of devices were selected for the particular operating systems. Also, there are frequent updates in the operating systems and browsers which might improve the performance of the web application. Similarly, there might be some error due to the use of an older version of the jQuery Mobile framework for the application development.

Figure 13 displays the web application available as a native application on the home screen of HTC desire running the Android 2.2 operating system.



Figure 13. Web application available at the home screen.

Figure 13 displays the mobile web application on the home page of the user's smartphone. This is done by using an application called OpenAppMkt. Currently this application can be downloaded only in iOS and Android with other platforms in the pipeline. OpenAppMkt allows the application written in HTML5 to be deployed in iOS and Android. Also, once the application has been placed in OpenAppMkt, it can be downloaded from both platforms. This makes it quite easy to open the web application as the users do not need to enter the full URL address and only click the application link at the home screen if they need to open the application. Other companies are sure to follow in this regard and create web application stores. Alternatively, the users can just enter the URL link or bookmark the web application in their browser and open it from there.

7 Discussion

Smartphones and tablets have multi-touch interfaces with which fingers are used as mouse pointers. Since the smartphones are smaller than desktops, fingers are not as precise as mouse pointers for doing some operations such as text selection. Fewer options are available using fingers than using the combination of keyboard and mouse in desktop computers. Similarly, the same desktop web page should not be used for rendering in mobile devices because the size aspect ratio is quite different. Thus, the users need to keep zooming in and out from the web page to use the application which results in bad user experience.

Today all the major platforms, iOS, Android and Windows Phone, take 30 percent of application revenue by selling the applications through their stores, and Apple also takes a 30-percent-cut on subscriptions sold through the App Store [25]. The developers get the remaining 70 percent of the revenue. Furthermore, money is wasted in getting developer accounts, and developers do not need permission to publish the applications developed with HTML5 since they can place application on the web in the form of a website.

Hence, this has led big companies such as Amazon, Financial Times and Vudu to develop HTML5 web applications instead of going through the market place of each platform. The developed web applications allow the companies to cut out different operating systems as the middle man, who would otherwise take a large part of the profit, and also to deploy all mobile platforms at once. HTML5 makes use of cache data, and thus the data usage for customers will decrease as well. [25]

One difficulty with using HTML5 and CSS3 is that there has not yet been a final specification for them, so the specifications might change in the future which would cause malfunctioning websites. Similarly, the support for HTML5 by various browsers is vague with different browsers supporting different features. The features supported on one browser may not be supported on another browser, so it is quite tedious and time consuming to test the application on various browsers. Along with this, the developers need to include features for the browsers not supporting HTML5 to give good user experience to the users.

Furthermore, the applications developed with HTML5 are typically slower than native applications. Thus, HTML5 cannot be used to develop games until it has matured fully and also differences in web browsers on different mobile platforms can raise the development costs. [25] Furthermore, the operating systems are free to choose which APIs to include in the next OS update along with new hardware devices. However, HTML5 support requires more compromise since there is various people involved. Thus, the development takes more time.

While I was developing the web application, the jQuery Mobile framework was still in the beta development stage. I had to use the beta 1 release version in the initial stage of my project. Developers from jQuery Mobile had released four alpha versions before that. The development committee there was releasing new versions each month. Hence, it was quite difficult to use the latest version. I have updated the application to use the first release candidate version of jQuery Mobile. At the time of this writing which is November 2011, the jQuery Mobile developers have just released the final stable release version 1.0. Nevertheless, development along with bug fixes is still in progress.

There were some difficulties in building the application as cross-platform. For examples, when I tried implementing the select menu option as a native function, it worked fine on iOS and desktops, but no menu popped up in Android devices. Then I had to use the custom menu by using the select menu as the pop up menu, and this worked in all the devices. Thus, some functions do not work as planned and the application should be tested properly. One possible reason for this could be that I had not used a stable release version of jQuery Mobile for the application development. The stable release version of jQuery Mobile might fix this problem.

Furthermore, lots of new features can be added in the application should the company decide to offer extra features. Possible features that can be added in the future include the possibility for the users to create an account just by taking a picture of their contract papers and by choosing the account credential to login. Pictures can already be taken on the Android operating system using HTML5 and flash, but iOS does not support flash. Thus, that solution is not universal. The developers can further improve

the application using PhoneGap to access various native features which will extend the application capabilities.

Similarly, the real time graph in the application can be developed further to show the graph for different hours. The database can be made to update itself every minute and the graph will be drawn accordingly when the users select the option. After that, the users can check their real time energy usage as well. This will be very helpful because the customers will then be able to see their energy usage in real time and determine which device is using more power. They can then either upgrade or change the devices consuming a lot of energy. Furthermore, when the customers are outside their home and they are not sure if they shut down the devices, they can check their energy consumption by opening the web application and make decisions based on that.

In addition to that, an online payment feature can also be added to the application. A verification method could be used as a web service in the application. Additionally, it could be made possible for the users to pay their energy bill through the application itself which would save the user the extra time of logging in other applications and doing the same.

8 Conclusion

The project focused on building a cross-platform web application for mobiles and tablets, and then studying how to make the application secure and cost-effective by using web services. The technology selected was the jQuery Mobile framework, HTML5, JavaScript, and WCF services. Also, PhoneGap was analyzed to offer native features to the web application should it need those in future. The web application was developed with future standards. The performance was still quite good on various browsers across different operating systems.

HTML5 has not yet been officially standardized and recommended by W3C. However, many of its features are supported by different browsers, and it is backwards compatible allowing old browsers to access it as well by limiting the new features. Similarly, jQuery Mobile and PhoneGap are open source frameworks which are developed actively. They are developed regularly to keep up with the new technologies. Developers can use them to develop cross-platform web applications in short time, and they can also ask help from the community should they ever have trouble with any features.

Similarly, WCF services can be used to increase the security of the web application. Using it, a company can use the same database for other client companies as well. The client companies have limited access to the database and so the database is more secure. Thus, the company with the database does not need to create an additional database for each client company and this saves money and resources.

Finally, the application was developed using the school server of the Helsinki Metropolia University of Applied Sciences, and so it has to be transported to the company's server for it to be used there. There has just been a stable release version for jQuery Mobile framework; however, the stable release version had not been used to develop the application. So, there might be some changes after the codes have been updates. The readers are suggested to use the latest version of jQuery Mobile and PhoneGap if they wish to develop applications using these technologies. For further information, they can also visit the official links of jQuery Mobile, PhoneGap and Microsoft to learn more about these technologies.

References

- 1 Allen Sarah, Graupera Vidal, Lundrigan Lee. Pro smartphone cross-platform development: iPhone, BlackBerry, Windows Mobile and Android development and distribution. USA: Apress Inc; 2010.
- 2 Brannan A. James, Ward Blake. iOS programming: A beginner's guide. USA: The McGraw-Hill Companies; 2011.
- 3 Pilone Dan, Pilone Tracey. Head first iPhone development. Sebastopol, CA: O'Reilly Media Inc; 2010.
- 4 Zhou Zhinan, Zhu Robert, Zheng Pei, Yang Baijian. Windows Phone 7 programming for Android and iOS developers. Indianapolis, Indiana: John Wiley & Sons Inc; 2011.
- 5 Jackson Wallace. Android apps for absolute beginners. USA: Apress Inc; 2011.
- 6 Google. What is Android? [online].
URL: <http://developer.android.com/guide/basics/what-is-android.html>.
Accessed 20 November 2011.
- 7 Ian Paul. After webOS, who's next To fall? [online]. PCWorld; 19 August, 2011.
URL: http://www.pcworld.com/article/238491/after_webos_whos_next_to_fall.
Accessed 20 November 2011.
- 8 Wargo M. John. BlackBerry development fundamentals. Boston: Pearson Education; 2010.
- 9 Kao Robert, Sarigumba Dante. BlackBerry for dummies. Indianapolis, Indiana: Wiley Publishing Inc; 2009.

- 10 Morris Ben, Bortenschlager Manfred, Luo Cheng, Lansdell Jon, Sommerville Michelle. Introduction to Bada: A developer's guide. West Sussex, UK: John Wiley & Sons Ltd; 2010.
- 11 Egham. Gartner says sales of mobile devices in second quarter of 2011 grew 16.5 Percent year-on-year; smartphones sales grew 74 percent [online]. UK: Gartner; 11 August, 2011.
URL: <http://www.gartner.com/it/page.jsp?id=1764714>.
Accessed 20 November 2011.
- 12 Basham Bryan, Sierra Kathy, Bates Bert. Head first servlets and JSP. Sebastopol, CA: O'Reilly Media Inc; 2008.
- 13 Apers Chris, Paterson Daniel. Beginning iPhone and iPad web apps. USA: Apress Inc; 2010.
- 14 Pilgrim Mark. HTML5 up and running. Sebastopol, CA: O'Reilly Media Inc; 2010.
- 15 Lunny Andrew. PhoneGap beginner's guide. Birmingham, UK: Packt Publishing; 2011.
- 16 Deep Blue Sky: Web design and consultancy. HTML5 and CSS3 support [online].
URL: <http://findmebyip.com/litmus>.
Accessed 20 November 2011.
- 17 Reid Jon. jQuery Mobile. Sebastopol, CA: O'Reilly Media Inc; 2011.
- 18 Bai Giulio. jQuery Mobile first look. Birmingham, UK: Packt Publishing; 2011.
- 19 Snell James, Tidwell Doug, Kulchenko Pavel. Programming web services with SOAP. Sebastopol, CA: O'Reilly & Associates Inc; 2002.
- 20 Abeysinghe Samisa. RESTful PHP web services. Birmingham, UK: Packt Publishing; 2008.

- 21 Löwy Juval. Programming WCF services, third edition. Sebastopol, CA: O'Reilly Media Inc; 2010.
- 22 Bustamante Leroux Michele. Learning WCF. Sebastopol, CA: O'Reilly Media; 2007.
- 23 Lecrenski Nick, Watson Karli, Ensor-Fonseca Robert. Beginning Windows Phone 7 application development: Building Windows Phone applications using Silverlight and XNA. Indianapolis, Indiana: Wiley Publishing Inc; 2011.
- 24 Gilmore Jason W. Beginning PHP and MySQL: From novice to professional, third edition. USA: Apress Inc; 2008.
- 25 Essers Loek. HTML5 adoption might hurt Apple's profit, research finds [online]. PCWorld; 12 September, 2011.
URL:http://www.pcworld.com/businesscenter/article/239867/html5_adoption_might_hurt_apples_profit_research_finds.html.
Accessed 3 November 2011.
- 26 Windows Phone. Execution model overview for Windows Phone [online]. Microsoft; 23 September 2011.
URL: [http://msdn.microsoft.com/en-us/library/ff817008\(v=VS.92\).aspx](http://msdn.microsoft.com/en-us/library/ff817008(v=VS.92).aspx).
Accessed 3 November 2011.

Appendices

Appendix 1: Ajax Navigation

Figure 14 displays the AJAX transition used to load an internal page in the application developed using the jQuery Mobile framework.

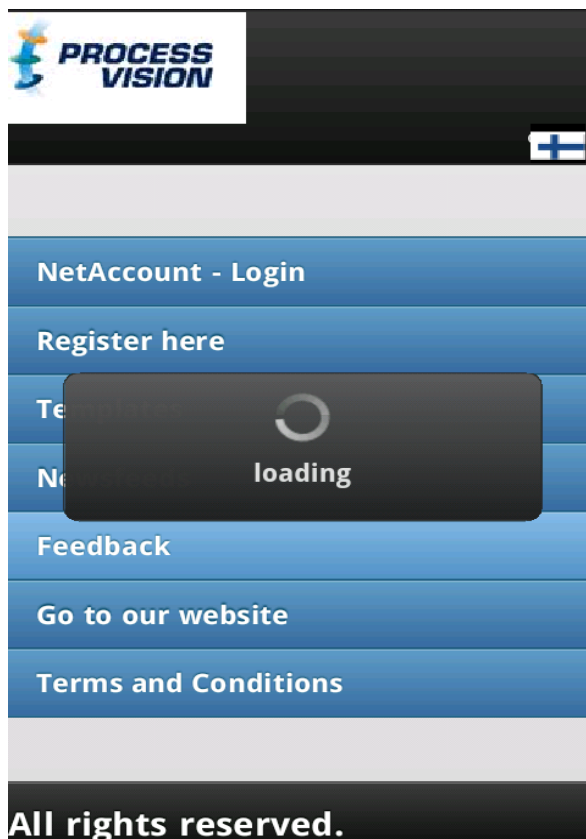


Figure 14. AJAX navigation with jQuery Mobile.

Figure 14 shows AJAX navigation in progress in the application. jQuery Mobile automatically uses Ajax navigation and if the developers need to specify a normal http request, they need to use the link attribute: rel= external. Ajax navigation can render the page fast.

Appendix 2: JavaScript Validation and Popup Page

Figure 15 shows the normal form and the message displayed with JavaScript when the user tries to submit an empty form.

The figure consists of two side-by-side screenshots of a mobile application interface. Both screenshots show a form titled "1. User Authentication" under the "PROCESS VISION" logo and a Finnish flag icon. The form instructs the user to "Insert the following fields that refers to the contract." and contains three input fields: "Customer Code", "Contract Code", and "Metering Point Code".

The left screenshot shows the normal form. The right screenshot shows the form after validation, with error messages "This field is required." displayed below each input field.

Figure 15. Normal form and form with error validated with JavaScript.

As can be seen in figure 15, the error message is displayed just below the form elements, so that the users can easily figure out where an error was made and solve the problem. If the device is held with a landscape view, then the error is displayed on the same line and on two lines if the error message does not fit in.

Figure 16 illustrates the popup page disappearing when the user opens a popup page and goes back to the previous page.

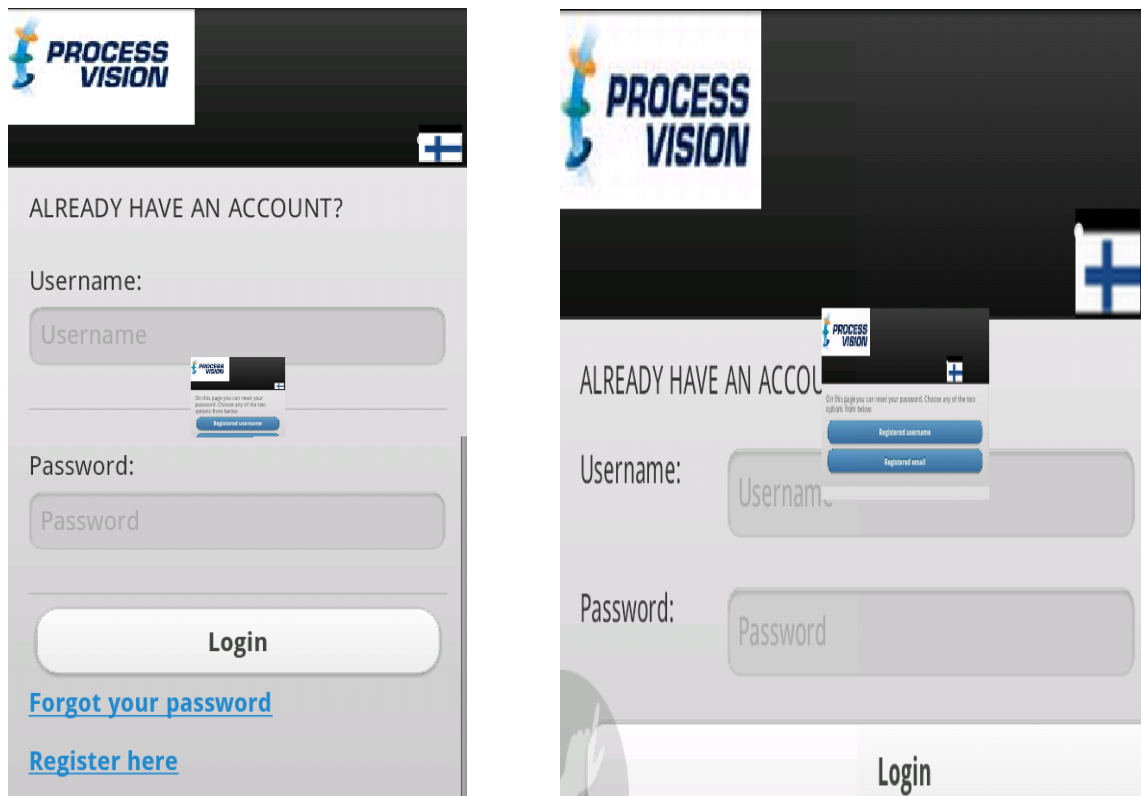


Figure 16. Popup page disappearing after pressing the back button.

Figure 16 above displays the popup page disappearing from the cache of the application in both landscape and portrait view of the application. The image on the left shows the view in portrait view while the right-hand side view shows the image in landscape view. Once the user opens up a popup page by clicking the link a new page opens up and when the user presses the back button again it disappears with a smooth transition using the jQuery Mobile framework.

Appendix 3: Custom Templates and Select Menu

Figure 17 displays the different custom templates available using the jQuery Mobile library.

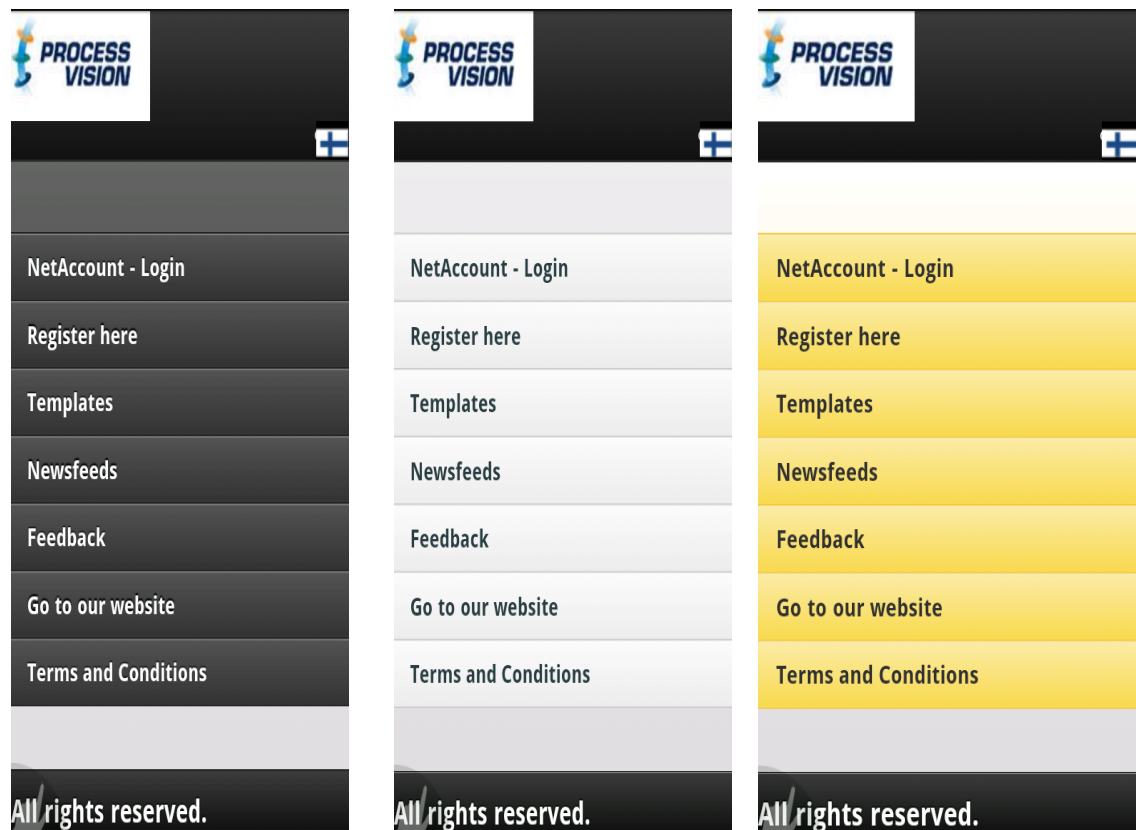


Figure 17. Custom templates available in jQuery Mobile.

The developers can make use of the custom templates available in jQuery Mobile library. If the developers need more templates they can edit the CSS file of jQuery Mobile and add more templates. Figure 17 shows three distinct templates available there. In addition to this, there is one more template as shown in figure 14, and the last template is white.

Figure 18 below shows the select menu option using the jQuery Mobile application.

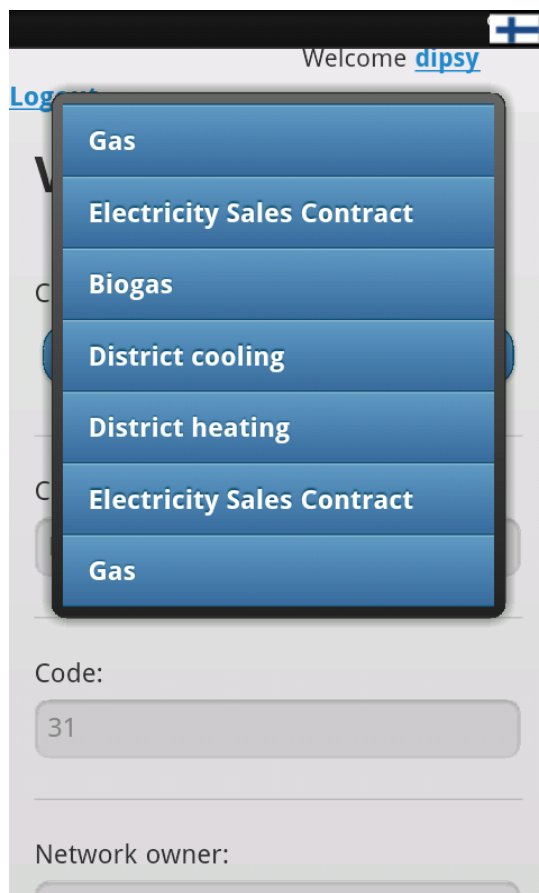


Figure 18. Using select menu in Android device.

The select menu option opens up a small new window on top of the present window as seen in figure 18. The figure shows a portrait view of the device. It is also possible to attach the select menu with the form but this feature is not supported by all the operating systems. Thus, a universal solution has been approached in this regard to support all the platforms by making the select menu appear in a new window.